Multiple Extremal Eigenpairs of Very Large Matrices by Monte Carlo Simulation

T. E. Booth

Applied Physics Division, Los Alamos National Laboratory, Los Alamos, NM 87545

J. E. Gubernatis

Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM 87545

(Dated: July 8, 2008)

We present a new Monte Carlo algorithm that allows the simultaneous determination of a few extremal eigenpairs of a very large matrix. It extends the power method and uses a new sampling method, the sewing method, that does a large state space sampling as a succession of samplings from a smaller state space. We illustrate the new algorithm by its determination of the two largest eigenvalues of the transfer matrix of a square Ising model at the critical temperature for sizes from 16×16 to 48×48.

A common problem in computational physics is computing the eigenpairs of large matrices. We will present a new Monte Carlo algorithm that allows the simultaneous determination of a few extremal eigenpairs of a very large matrix without the need to orthogonalize pairs of vectors to each other or store all the components of any vector.

The new algorithm is an extension of the power (projection) method [1], which is the traditional starting point for a Monte Carlo determination of the eigenpair associated with the eigenvalues of largest absolute value λ_1. While various Monte Carlo versions of the power method often compute this dominant eigenvalue very well, computing subdominant eigenvalues $\lambda_2, \lambda_3, \ldots$ has often proven much more difficult and is much less frequently attempted [2]. Our Monte Carlo power method computes multiple extremal eigenpairs simultaneously and straightforwardly. For the particular algorithm presented we also introduce a new sampling method, the sewing method, that does a large state space sampling as a succession of small state space samplings. Although our new algorithm is extendable to finding more than two extremal eigenpairs, here we will focus on finding just λ_1 and λ_2.

Our ultimate targets are matrices so large that they are unassailable deterministically because no single vector can be stored in memory. As the system size increases, finding a few extremal eigenpairs of the transfer matrix of the two-dimensional Ising model becomes such a problem. Its two extremal eigenvalues are also of significant physical interest: the logarithm of λ_1 is proportional to the free energy, and the ratio λ_2/λ_1 controls long range spin correlations near the critical point [3, 4]. Onsager [4] derived exact expressions the eigenvalues of this transfer matrix for any finite-sized system thereby providing a nearly unprecedented opportunity for non-trivially benchmarking our algorithm for exceptionally large systems. We comment however that there is no *a priori* restriction of our algorithm to problems in classical statistical mechanics. It is also applicable to transfer matrices of quantum origin and to ground and low lying excited states of many Hamiltonian matrices. In addition it has application to more widely diverse problems as the nuclear critically problem [5]. We also note that our extension of the power method is not limited to a Monte Carlo implementation [6].

The transfer matrix $A(\sigma, \sigma')$ of an $m \times m$ Ising model with periodic boundary conditions in one direction and open boundary conditions in the other is a $2^m \times 2^m$ matrix whose elements are [3]

$$A\left(\sigma, \sigma'\right) = \exp\left(\nu \sum_{k=1}^{m-1} \mu_k \mu_{k+1}\right) \exp\left(\nu \sum_{k=1}^{m} \mu_k \mu_k'\right) \quad (1)$$

with $\nu = J/k_B T$, J is the exchange constant, k_B being Boltzmann's constant, and T equal to the temperature. The Ising spin variable μ_k has the value of ± 1, $\mu_{m+1} = \mu_1$, and the symbol $\sigma = (\mu_1, \mu_2, \ldots, \mu_m)$ denotes a configuration of Ising spins. (There are 2^m possible configurations .) Numerically, we represented a σ by the first m bits of integers ranging from 0 to $2^m - 1$. We comment that all the elements of $A(\sigma, \sigma')$ are greater than zero so the matrix is maximally dense, and because of the one open boundary, it is also asymmetric. From the Perron-Frobenius Theorem [7] we have that a dominant eigenvalue that is real and positive. Further, all components of the corresponding eigenstate are real and have the same sign.

The power method [1] for some real-valued $M \times M$ matrix A, not necessarily symmetric, is an iterative procedure, started with some ψ, normalized in a convenient but otherwise relatively arbitrary, manner, that cycles the two steps

$$\begin{aligned} \phi &= A\psi \\ \psi &= \phi/\|\phi\| \end{aligned} \quad (2)$$

until some convergence criterion is met. If (λ_i, ψ_i) are eigenpairs of A, then starting with some

$$\psi = \sum_{\alpha=1}^{M} \omega_\alpha \psi_\alpha$$

and specifying $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_N|$, we find after n iterations that

$$A^n \psi = \lambda_1^n \left[\omega_1 \psi_1 + \sum_{\alpha=2}^{M} \omega_\alpha \left(\frac{\lambda_\alpha}{\lambda_1} \right)^n \psi_\alpha \right] \qquad (3)$$

Accordingly, $\psi \to \psi_1/\|\psi_1\|$ and $\|\phi\| \to \lambda_1$ as $n \to \infty$.

Finding more than one eigenpair by the power method requires initializing the method with more than one starting point [1, 8]. For methods of which we are aware, these starting points need to be orthogonal, and this orthogonality needs to be maintained, at least periodically, throughout the iteration. This is much more difficult to do in a Monte Carlo procedure than in a determinisitic one [2].

In developing a Monte Carlo algorithm to estimate two extremal eigenvalues, we will exploit several observations of Booth [9, 10]. He noted that for any eigenpair (λ, ψ) and for each non-zero *component* of the eigenvector, the eigenvalue equation $A\psi = \lambda\psi$ can be rewritten as

$$\lambda = \frac{\sum_j A_{ij} \psi_j}{\psi_i} \qquad (4)$$

and that similar equations can also be written for any number of groupings of components,

$$\lambda = \frac{\sum_{i \in R_1} \sum_j A_{ij} \psi_j}{\sum_{i \in R_1} \psi_i} = \frac{\sum_{i \in R_2} \sum_j A_{ij} \psi_j}{\sum_{i \in R_2} \psi_i} = \cdots = \frac{\sum_{i \in R_L} \sum_j A_{ij} \psi_j}{\sum_{i \in R_L} \psi_i} \qquad (5)$$

where the R_i are rules for different groupings. Here, we will exploit the fact that any two groupings, say 1 and 2, imply

$$\sum_{i \in R_2} \psi_i \sum_{i \in R_1} \sum_j A_{ij} \psi_j = \sum_{i \in R_1} \psi_i \sum_{i \in R2} \sum_j A_{ij} \psi_j \qquad (6)$$

This directly follows from (5).

As do standard procedures for finding for just the two extremal eigenvalues, we also use two normalized, starting points $\psi' = \sum_\alpha \omega'_\alpha \psi_\alpha$ and $\psi'' = \sum_\alpha \omega'_\alpha \psi_\alpha$ but they need not necessarily be orthogonal [6, 9, 10]. At each step of the power method, we apply A to them individually; however, to prevent both from projecting to the same dominant eigenfunction, we adjust at each step the relationship between their *sum* to direct one to the dominant state and the other to the next dominant one. We do this in the following way: we start the iteration with $\psi = \psi' + \eta\psi''$. If at the nth step, ψ' and ψ'' have iterated to $\hat{\psi}'$ and $\hat{\psi}''$, then at the $(n+1)^{th}$ step, we use (5) and (6)

$$\frac{\sum_{i \in R_1} \sum_j A_{ij} \hat{\psi}'_j + \eta \sum_{i \in R_1} \sum_j A_{ij} \hat{\psi}''_j}{\sum_{i \in R_1} \hat{\psi}'_i + \eta \sum_{i \in R_1} \hat{\psi}''_i} \qquad (7)$$

$$= \frac{\sum_{i \in R_2} \sum_j A_{ij} \hat{\psi}'_j + \eta \sum_{i \in R_2} \sum_j A_{ij} \hat{\psi}''_j}{\sum_{i \in R_2} \hat{\psi}'_i + \eta \sum_{i \in R_2} \hat{\psi}''_i}$$

to obtain

$$q_2 \eta^2 + q_1 \eta + q_0 = 0 \qquad (8)$$

The algorithm thus is to apply A repeatedly to ψ' and ψ''. If two real solutions η_1 and η_2 of (8) exist, then update via

$$\begin{aligned} \psi' &\leftarrow A\psi' + \eta_1 A\psi'' \\ \psi'' &\leftarrow A\psi' + \eta_2 A\psi'' \end{aligned} \qquad (9)$$

otherwise via

$$\begin{aligned} \psi' &\leftarrow A\psi' \\ \psi'' &\leftarrow A\psi'' \end{aligned} \qquad (10)$$

After the η's become real, η_1 guides further iterations to (λ_1, ψ_1); η_2, to (λ_2, ψ_2). The eigenvalues are estimated from

$$\begin{aligned} \lambda_1 &= \frac{\sum_{i \in R_1} \sum_j A_{ij} \psi'_j + \eta_1 \sum_{i \in R_1} \sum_j A_{ij} \psi''_j}{\sum_{i \in R_1} \psi'_i + \eta_1 \sum_{i \in R_1} \psi''_i} \\ \lambda_2 &= \frac{\sum_{i \in R_1} \sum_j A_{ij} \psi'_j + \eta_2 \sum_{i \in R_1} \sum_j A_{ij} \psi''_j}{\sum_{i \in R_1} \psi'_i + \eta_2 \sum_{i \in R_1} \psi''_i} \end{aligned} \qquad (11)$$

where η_1 and η_2 generate the largest and next largest eigenvalue estimates. A justification of this procedure is given in [6].

The Monte Carlo method is used to estimate the result of the repeated matrix-vector multiplication. In the basis defining the matrix elements of A, we write

$$\begin{aligned} \psi' &= \sum_i \omega'_i |i\rangle \\ \psi'' &= \sum_i \omega''_i |i\rangle \end{aligned}$$

and call the amplitudes ω'_i and ω''_i *weights* even though they are not necessarily all positive nor are the sums of their absolute values unity. We assume that the elements of the $M \times M$ matrix A are easily generated on-the-fly as opposed to being stored. Next, we imagine we have N particles distributed over the M basis states and interpret A_{ij} as the weight of particles arriving in state $|i\rangle$ on iteration $n+1$ per unit weight of a particle in state $|j\rangle$ on iteration n and will regard the action of A on a ψ as causing a particle to jump from some $|j\rangle$ to some $|i\rangle$, carrying its current weight ω_j, modified by A_{ij}, to state $|i\rangle$. To do this, we define the total weight leaving state $|j\rangle$ as

$$W_j = \sum_i A_{ij} \qquad (12)$$

and the transition probability from $|j\rangle$ to $|i\rangle$ as

$$T_{ij} = A_{ij}/W_j \qquad (13)$$

Instead of always (i.e., with probability 1) moving weight A_{ij} from state $|j\rangle$ to state $|i\rangle$, we will instead sample a $|i\rangle$ from T_{ij} and multiply the transferred weight by W_j

For many Monte Carlo simulations, as is the case for the transfer matrix of the Ising model, the particle weights defining the eigenvector associated with the largest eigenvalue can be made all positive. The second eigenfunction however must be represented by some particles of negative weight and some particles of positive weight. For some jumps these negative and positive weights must at least partially cancel to maintain a correct estimation of the second eigenfunction. When $N \ll M$, as is typical, this cancellation does not occur often enough in a Monte Carlo simulation without proper design.

There are several ways to design the cancellation [9]. For our Ising simulations, we promoted this cancellation by sorting the particles into state order (a state is represented by the bits of an integer) at the end of each iteration. Particles 1 and 2 are then sampled together according to the Arnow *et al.* scheme [11], then particles 3 and 4, and so forth. An ordered list means there are (typically) many nearby states $|i\rangle$ accessible from both particles ℓ and $\ell + 1$ with nontrivial transition probabilities.

As the iteration progresses, the absolute value of the weights of some particles becomes very large, and those of some others, very small. As standard for Monte Carlo methods with weighted particles, we stochastically eliminated particles with weights of small magnitude and stochastically split those with large magnitudes. To do this we used a procedure called the comb [12].

The steps of the algorithm are: First, we initialize the weights of two vectors. For the Ising simulation, we selected ω_i' uniformly and randomly over the interval $(0,1)$ and the ω_i'' uniformly and randomly over the interval (-1,1). Then, for a fixed number of times we iterate. For each iteration we execute the jump procedure for each particle, place the particle list in state order, effect cancellations, estimate the eigenvalues from (11), update ψ' and ψ'', and then comb.

If M is sufficiently small so we can store all components of our vectors, sampling from the cumulative probability $C_i = \sum_{k=0}^{i} T_{kj}$ works well. If the number of states gets too large (e.g., $m > 12$), then C_i cannot be sampled directly because it cannot fit in the computer's memory. In this case, we could just randomly pick from any state $|j\rangle$ any state $|i\rangle$ with probability $1/M$ instead of always picking a state $|i\rangle$ (i.e., with probability 1). The problem with this approach is that the A_{ij} can have immense variation so that this simple sampling scheme is unlikely to work well as a Monte Carlo method. This situation is especially true for the Ising problem. A large part of

such variations however can be removed by sampling the new state in stages and then sewing the stages together.

To explain our sewing procedure, we will first assume that we can write any state $|i\rangle$ in our basis as a direct product of the states in a smaller basis, $|i\rangle = |i_2\rangle |i_1\rangle$. Instead of transferring weight W_j from state $|j\rangle$ to state $|i\rangle$ with probability T_{ij}, we will use the a_{ij} that would apply to the smaller set of states and then make an appropriate weight correction. For the Ising model, a_{ij} is the transfer matrix of a smaller lattice size.

For each smaller set of states, we rewrite the analogous transition probability from state $|j\rangle$ to state $|i\rangle$ as

$$t_{ij} = a_{ij}/w_j, \qquad (14)$$

and the analogous weight multiplier as

$$w_j = \sum_k a_{kj} \qquad (15)$$

We thus will sample $|i_1\rangle$ and $|i_2\rangle$ from the probability function

$$t(i_2, j_2) t(i_1, j_1) \qquad (16)$$

The weight correction C_{ij}, necessary to preserve the expected weight transfer from state $|j\rangle$ to $|i\rangle$, satisfies

$$A_{ij} = C_{ij} t(i_2, j_2) t(i_1, j_1) \qquad (17)$$

Thus

$$C_{ij} = w_{j_1} w_{j_2} \frac{A_{ij}}{a(i_1, j_1) a(i_2, j_2)} \qquad (18)$$

The sewing method generalizes easily. For k sets of states, (17) and (18) become

$$A_{ij} = C_{ij} \prod_{n=1}^{k} t(i_n, j_n) \qquad (19)$$

with

$$C_{ij} = A_{ij} \prod_{n=1}^{k} \frac{w_{j_n}}{a(i_n, j_n)} \qquad (20)$$

For the Ising problem, we took for state $|i_1\rangle$ first m/k bits of the integer representing $|i\rangle$; for $|i_2\rangle$, the second set; etc. The weight correction becomes

$$C_{ij} = \exp(\nu D_i) \prod_{n=1}^{k} w_{j_n} \qquad (21)$$

where apart form a factor of ν, D_i is the energy difference between calculating with the bits together and the bits separately. It is straightforward to calcualte.

Using this sewing algorithm for the sampling of states, we computed the first and second eigenvalues for $m = 16$

TABLE I: Our Monte Carlo power method estimates of λ_1 and λ_2 and their statistical errors for variously sized square Ising models. Each estimate was based on 20 independent simulations. Also given are the values computed via Onsager's exact results [4].

Matrix Size	λ_1 (Onsager)	λ_1	λ_2 (Onsager)	λ_2
$2^{16} \times 2^{16}$	2.93297×10^6	$2.93307 \pm 0.00008 \times 10^6$	2.79225×10^6	$2.79482 \pm 0.00010 \times 10^6$
$2^{32} \times 2^{32}$	8.39316×10^{12}	$8.39311 \pm 0.00049 \times 10^{12}$	8.18959×10^{13}	$8.18807 \pm 0.00061 \times 10^{12}$
$2^{48} \times 2^{48}$	2.41504×10^{19}	$2.41522 \pm 0.00019 \times 10^{19}$	2.37584×10^{19}	$2.37481 \pm 0.00054 \times 10^{19}$

to $m = 48$ Ising models by sewing 6 sets of 8 bits. We note that $2^{48} \approx 2.8 \times 10^{14}$. The results are shown in Table I. To compute averages and standard errors for each size, we executed twenty independent Monte Carlo runs, each with 1 million particles per iteration (5 million for $m = 48$), 500 iterations per run and used only the second half of the iterations in each run for the estimation process. $\nu = 0.4406867935097715$ [3, 4], the value at the critical temperature. Presented are 3 systems sizes. For each we give the values of λ_1 and λ_2 predicted from Onsager's expression and from our enhanced power method. We see that our Monte Carlo produced eigenvalues agree very well with Onsager's predictions. For our eigenvalue estimates, R_1 consisted of the states for which more than half of its m bits were 0's and R_2 consisted of the states for which more than half of its m bits were 1's. More results and algorithmic detals will be given elsewhere [13].

We anticipate the sewing algorithm being applicable to other many-body problems defined on a lattice. Likely, these applications will require more sophisticated programming than for the Ising model. The "sewing" method for the Ising model worked well as high as $m = 60$, sewing together 6 sets of 10 bits. For $m > 60$, our computer codes would need significant modification to implement a more flexible scalable procedure for representing a state configuration requiring more than a single computer word. We do not know how large an m can be accommodated with a better computer program.

The transfer matrix of the Ising model is real, positive, asymmetric, and dense. How is our algorithm changed if a matrix lacks one or more of these properties? For simple test cases, we have successfully constructed deterministic procedures for matrices whose elements are complex valued. Also, in this context, we have had success for real asymmetric matrices whose eigenvalues are complex valued. Devising Monte Carlo algorithms for real, symmetric, sparse matrices has however received more of our attention [14]. To find the eigenvalue of smallest size, if it not the one with the largest absolute value, one simply uses a shifted matrix, $A \to A - \sigma I$.

In closing, we believe that our new algorithm is accurate, easy to implement, and applicable to many other problems. Wider use of the algorithm will define more crisply its strengths and limitations than is possible by just the present application. The intent of the present application was benchmarking and not studying the scaling of the eigenvalues of the transfer matrix of the two-dimensional Ising model. Both deterministic and Monte Carlo power methods have been used for such studies. Deterministic methods [15] have computed λ_1 and λ_2, while Monte Carlo methods [16], just λ_1. The system sizes were considerably smaller ($m \leq 25$ deterministically and $m \leq 21$ stochastically) than the largest size ($m = 48$) presented here. This size should not be the largest accessible by our methods. All our calculations were done on a single processor.

[1] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, (Oxford University Press, Oxford, 1965).
[2] For example, B. L. Hammond, W. A. Lester, Jr., and P. J. Reynolds, *Monte Carlo Methods in Ab Initio Quantum Chemistry*, (World Scientific, Singapore, 1994).
[3] C. J. Thompson, *Mathematical Statisical Mechanics*, (Princeton University Press, Princeton, 1972).
[4] L. Onsager, Phys. Rev. **65**, 117 (1944); B. Kaufman, Phys. Rev. **76**, 1232 (1949) .
[5] Iván Lux and László Koblinger, *Monte Carlo Particle Transport Methods: Neutron and Photon Calculations*, (CRC Press, Boca Raton, 1991).
[6] J. E. Gubernatis and T. E. Booth, J. Comp. Phys., to appear, arXiv:0807.1261v1.
[7] H. S. Wilf, *Mathematics for the Physical Sciences*, (Dover, New York, 1978).
[8] G. H. Golub and C. F. van Loan, *Matrix Computations*, (Johns Hopkins University Press, Baltimore, 1989).
[9] T. E. Booth, Nucl. Sci. Eng. **143**, 291 (2003).
[10] T. E. Booth, Nucl. Sci. Eng. **154**, 4862 (2006).
[11] D. M. Arnow *et al.*, J. Chem. Phys. **77**, 5562 (1982).
[12] Donald H. Davis, Lawrence Radiation Laboratory Report UCRL-6707.
[13] T. E. Booth and J. E. Gubernatis, Los Alamos National Laboratory Report, LA-UR-07-7672.
[14] J. E. Gubernatis and T. E. Booth, unpublished.
[15] For example, H. L. Richards, M. A. Novotny, P. A. Rikvold, Phys. Rev. B **48**, 14584 (1993).
[16] For example, M. P. Nightingale and H. W. J. Blöte, Phys. Rev. B **54**, 1001 (1996) and J. Kaupužs, Int. J. Mod. Physics. C, **17**, 1095 (2006).